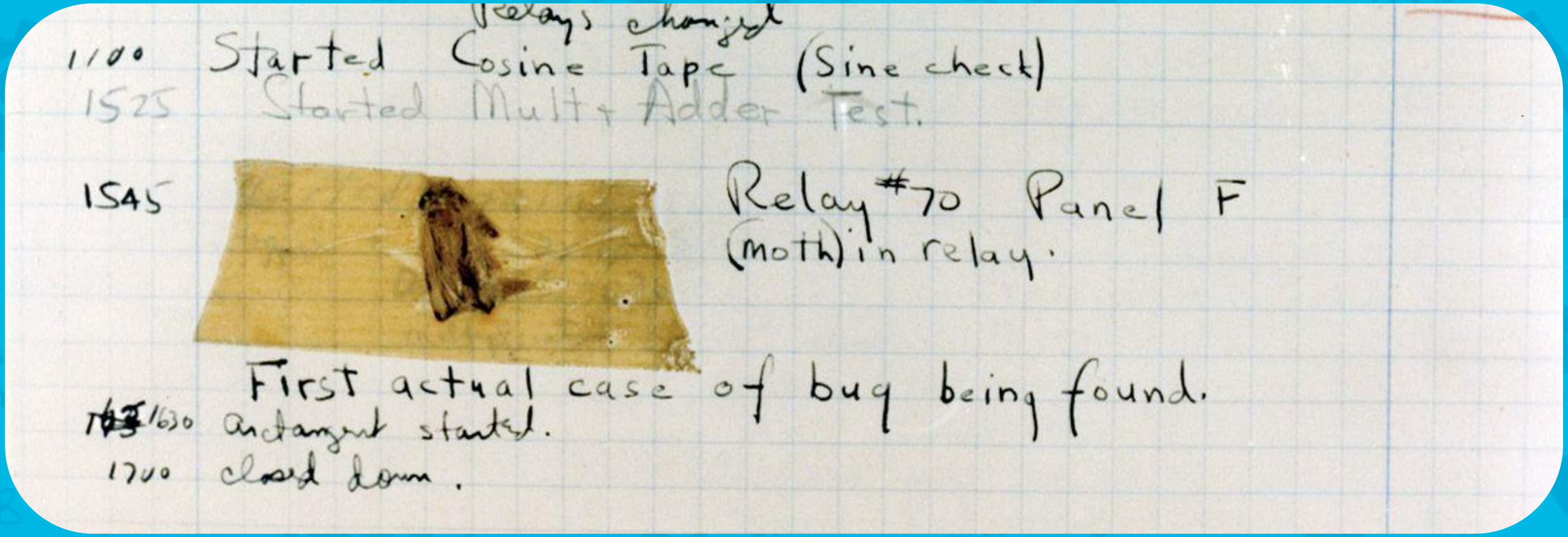
Measurements, Sensors and Data Logging Course

Week 5

Upcoming Weeks

- Office Hours
  - Monday Dec. 14th @ 7:00 PM

- Weekly Sessions
  - Thursday Dec. 17th @ 7:00 PM

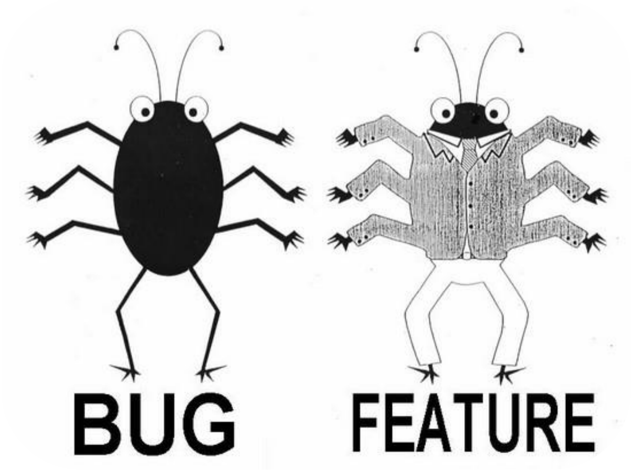


Relays changed
1100 Started Cosine Tape (Sine check)
1525 Started Mult+ Adder Test.
1545 Relay #70 Panel F
(moth) in relay.
First actual case of bug being found.
1630 Antangent started.
1700 closed down.

Debugging!

Debugging

- Steps
  - Attempt to Upload and view in serial monitor: *DebugProgram_5*
  - Desired Outcome: read temperature and humidity values



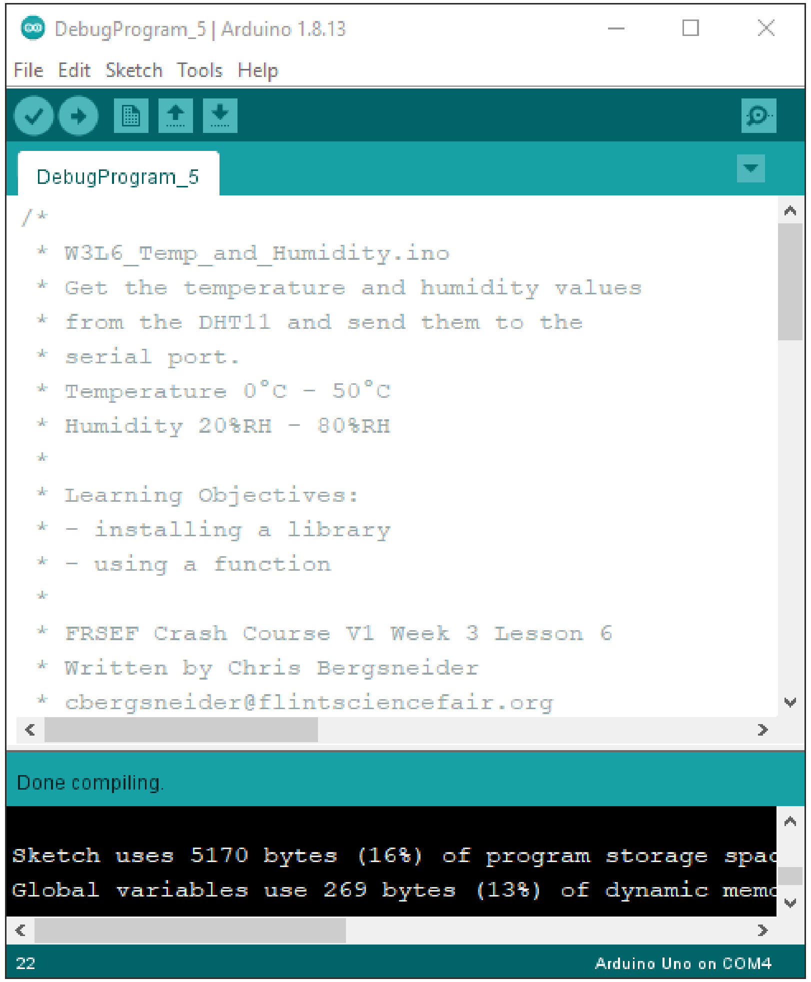
Debugging

- Solution
  - *DebugProgram_5*

incorrect pin for the temperature sensor in sketch
- D3 is the correct pin for the Temp & Humidity sensor

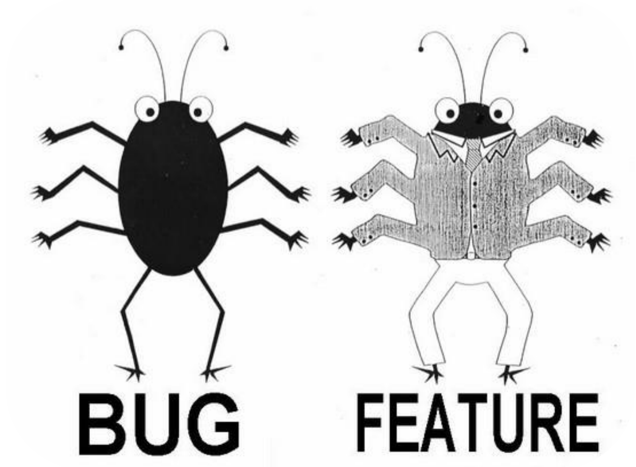
~~`const byte dht11Pin = 6;`~~ (incorrect line)

`const byte dht11Pin = 3;`



Debugging

- Steps
  - Attempt to Upload and view in serial monitor: *DebugProgram_6*
  - Desired Outcome: read temperature and humidity values



Debugging

- Solution
  - *DebugProgram_6*

Variable not declared in sketch

- `temp` *was meant to be* `tempC`

~~`Serial.print(temp);`~~ (incorrect line)

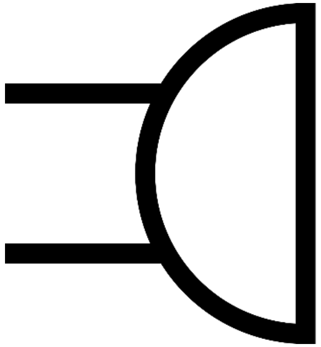
`serial.print(tempC);`

Sensors & Applications

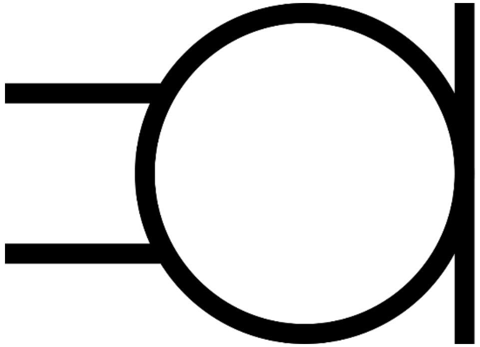
Sensors & Applications: Microphone

- What is a microphone?
  - A microphone is a transducer that converts sound wave to an electrical signal.
  - Microphones are used to record music and voice, but also used for scientific analysis.
- Where are microphones used?
  - Audio recording, cell phones, walkie-talkie, computers, sonar, presence detection, knock detection, etc.
- How do we use the Microphone?
  - Microphones must be amplified or conditioned before we can use the signal. We can then read the analog signal with the ADC in the microcontroller.
- More Information:
  - https://en.wikipedia.org/wiki/Microphone

Microphone Electrical Symbols



By ErikBuer - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=50257723



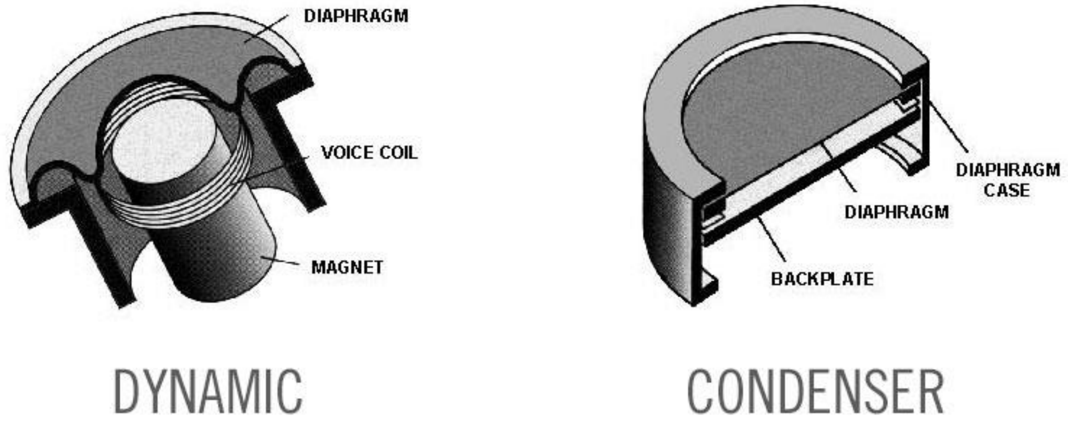
By ErikBuer - Own work, CC BY-SA 4.0, https://commons.wikimedia.org/w/index.php?curid=50257685

Microphone

Downloaded from https://www.digikey.com/en/products/detail/cui-devices/CMA-6542PF/1869980

Sensors & Applications: Microphone

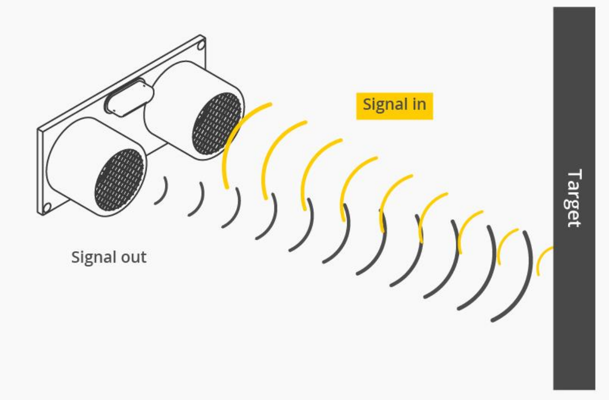
- There are multiple types of microphones, most common are dynamic and condenser



- More Information:
  - https://taylor-sound.com/taylor-sound-blog/whats-the-difference-between-a-dynamic-and-condenser-microphone/

Sensors & Applications: Ultrasonic Sensors

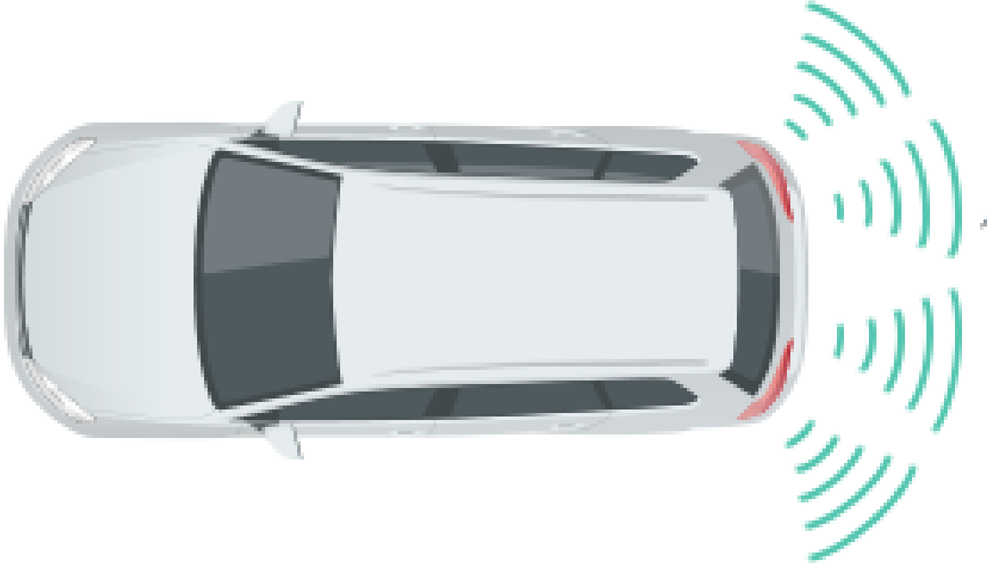
- Application using microphones: ultrasonic sensors
  - Ultrasonic Sensors
    - Simple range finders
    - Object position and tracking
      - Example: automotive active safety



- https://www.seeedstudio.com/blog/2020/01/03/what-is-a-sound-sensor-uses-arduino-guide-projects/#:~:text=A%20sound%20sensor%20is%20defined,converting%20it%20to%20electrical%20signals

Sensors & Applications: Ultrasonic Sensors

- Ultrasonic Sensors
  - Object position and tracking
    - Example: automotive active safety



Sensors & Applications: Microphone

At Home Activity

1. Open Microphone Sketch
   a. **File → Sketchbook → FRSEF_Crash_Course → Week_2 → W2L5_Microphone.ino**
2. Verify the sketch by clicking the Verify Button.
   a. The sketch should compile with no errors.
3. Upload the sketch to your Arduino by clicking the Upload Button.
   a. The sketch should re-compile, and then upload to your Arduino.
4. Open the serial monitor.
   a. **Tools → Serial Plotter** (Ctrl+Shift+L)
5. Observe the output in the Serial Plotter

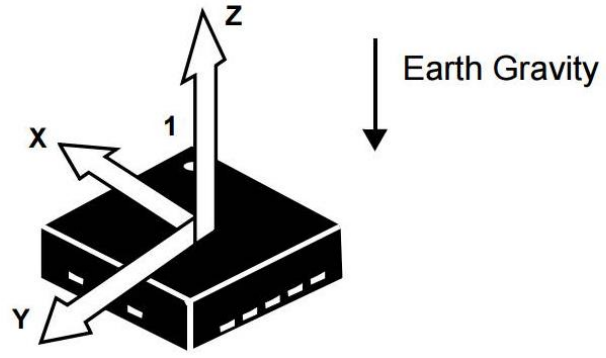
Accelerometer

Lesson 10

Motion Sensors

Accelerometer

- Accelerometers are used to measure acceleration, in linear directions
- Measurements in m/s^2
  - $1g$ = 9.8 m/s^2
  - $0g$ = object not moving or in free fall
- Types: analog, digital (IIC, SPI), PWM
- Uses
  - Position tracking
  - Force measurement
  - Vibration measurement

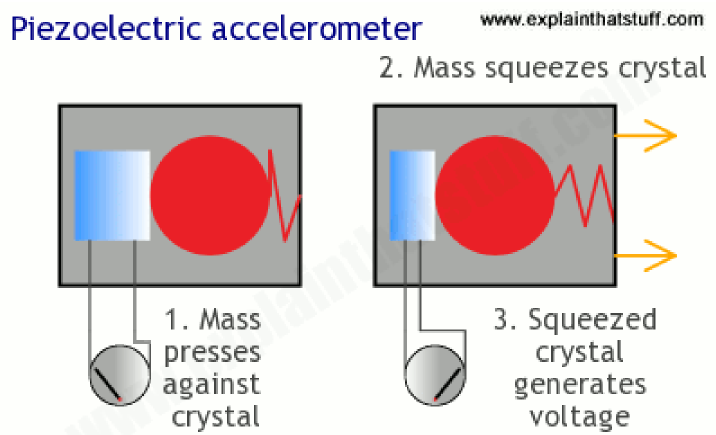


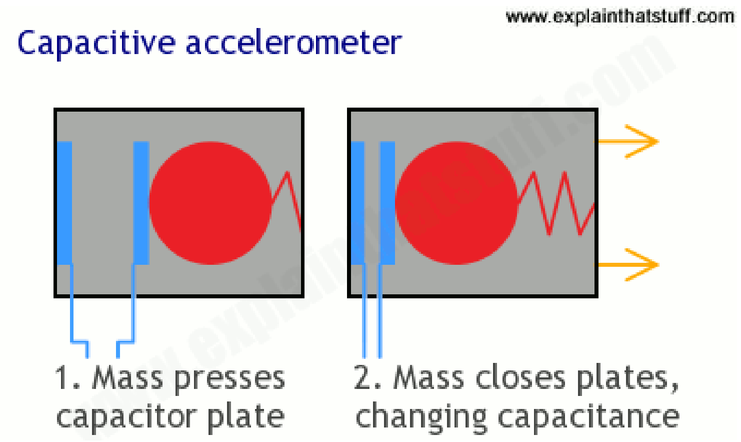
(TOP VIEW)
DIRECTION OF THE
DETECTABLE ACCELERATIONS

- https://www.adafruit.com/category/521
- https://learn.sparkfun.com/tutorials/accelerometer-basics/all
- https://www.sparkfun.com/pages/accel_gyro_guide?_ga=2.260802829.124947883.1606791953-761679650.1605223049
- https://www.seeedstudio.com/blog/2019/12/24/what-is-accelerometer-gyroscope-and-how-to-pick-one/

Motion Sensors
Accelerometer

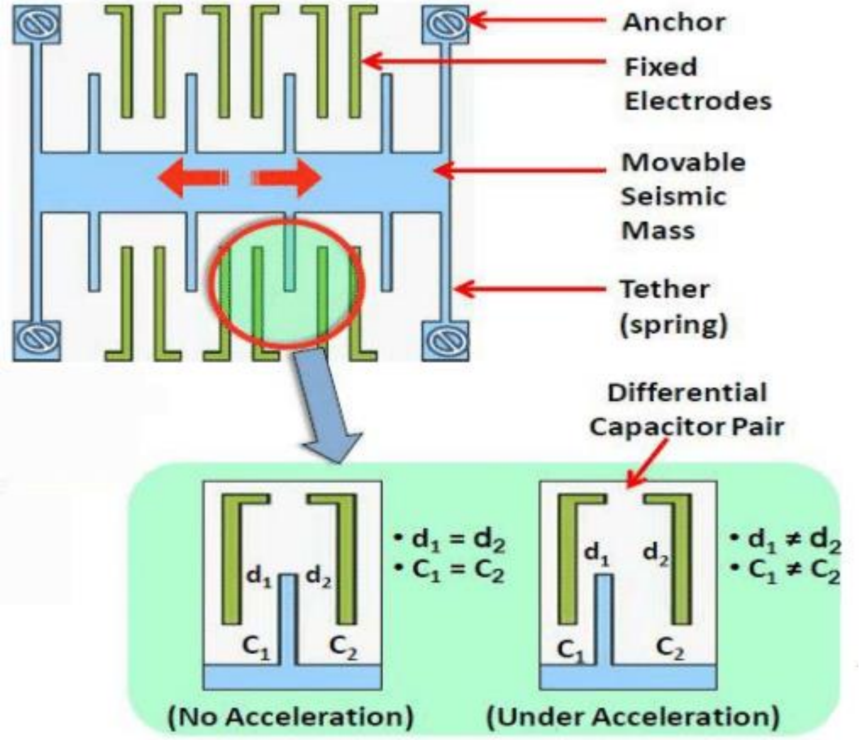
- Accelerometers are used to measure acceleration, in linear directions
- How they work
  - Capacitive: capacitive plates internally, some fixed and others on springs, motion between plates causes change in capacitance
  - Piezo Electric: Small mass on springs around piezo-electric materials. Electrical charges are created.





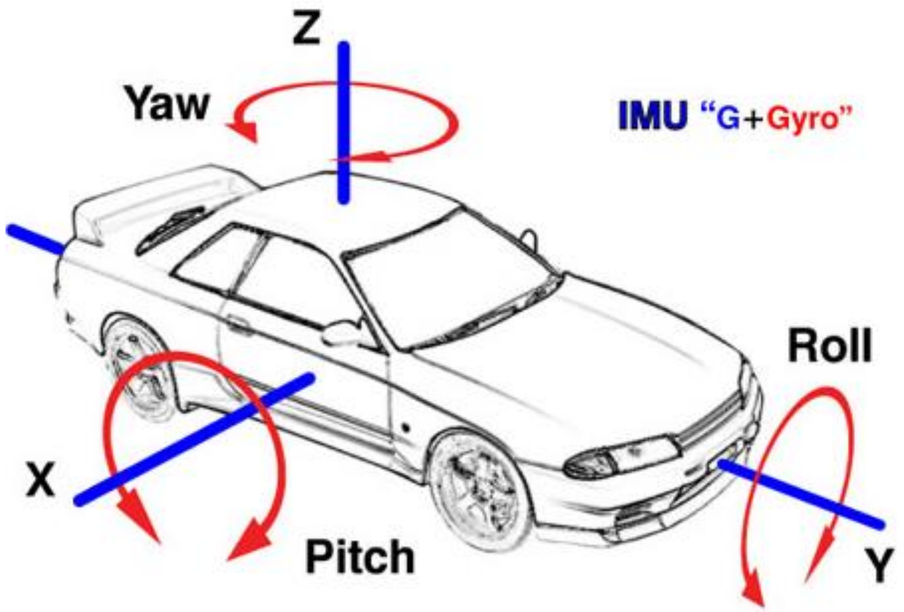
Motion Sensors
Accelerometer

- How they work
  - MEMS: microscopic, silicon based moving mass
    - Uses either piezo or capacitive changes
    - Cool Graphic



Motion Sensors
Accelerometer

- Other commonly used motion sensors:
  - Gyroscope: measure rotational motion
  - Magnetometer: measure magnetic force, typically magnetic north
- IMU: Accelerometer + Gyroscope



Hardware
Accelerometer

- What hardware will we need for this Lesson?
  - Grove 3-axis Accelerometer Module on IIC
  - Seeeduino Lotus (Arduino Uno compatible board)
  - SD Card Shield + SD Card
- Please assemble parts the same way we did last week

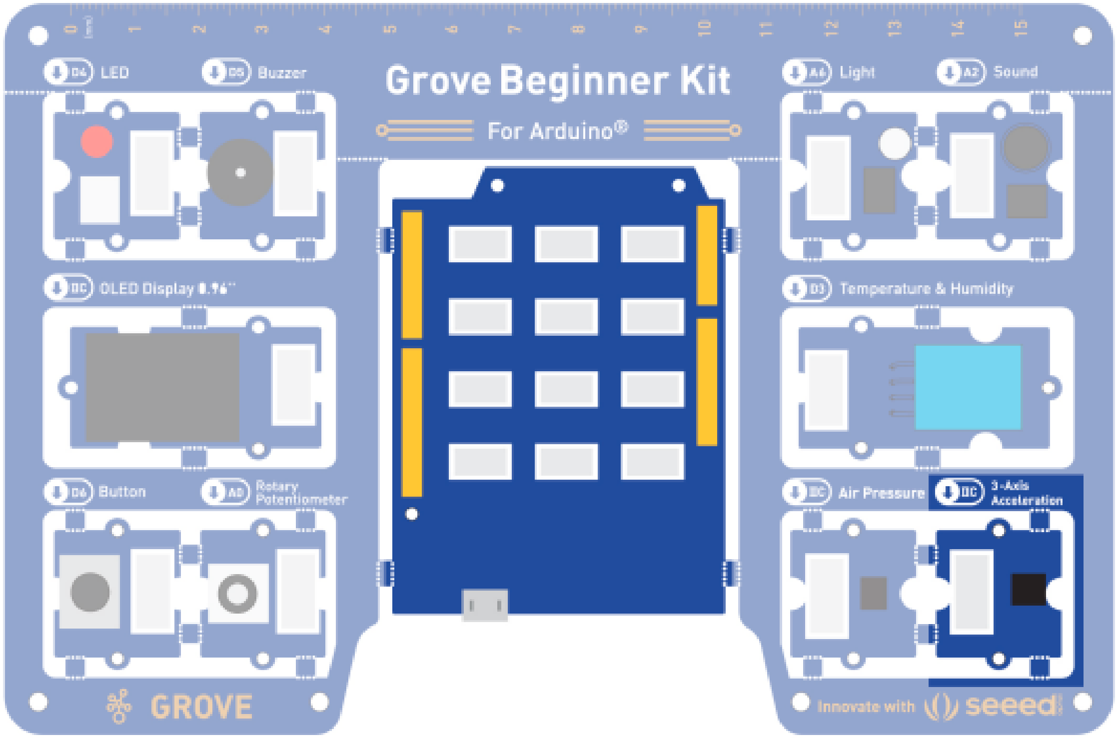


Image copied from https://www.amazon.com/HiLetgo-Logging-Recorder-Logger-Arduino/dp/B00PI6TQWO/

Image copied from https://www.microcenter.com/product/485234/micro-center-64gb-microsdxc-class-10--uhs-1-flash-memory-card

Library

Accelerometer

- Library to use: *sparkfun LIS3DH*
  - Search for ***Sparkfun LIS3DH*** in the library manager and install it.
  - There are multiple variants available for the LIS3DH sensor.
  - `#include <SparkFunLIS3DH.h>`
  - `LIS3DH myIMU(I2C_MODE, 0x19);`
- More Information:
  - https://github.com/sparkfun/SparkFun_LIS3DH_Arduino_Library

Open and Upload Sketch

Lesson 10: Accelerometer

1. Open Simple Datalogger Sketch
   - **File → Sketchbook → FRSEF_Crash_Course → Week_5 → L10_Accelerometer.ino**
2. Upload the sketch to your Arduino by clicking the Upload Button.
   - The sketch should compile, and then upload to your Arduino, assuming you have the correct
3. Move the board around so that different faces point down for a couple seconds then gently shake or drop your board (from a couple inches)
4. Unplug the USB connector
5. Remove SD card and plug into your computer
6. Look at the data in your spreadsheet program
   - note the name of the file is now a number repressing the day, hour, minutes and seconds that the log started at. This prevents us from overwriting or appending data to older files!

Activity
Accelerometer

- With SD Card shield attached, start recording and conduct multiple drop tests.
  - Conduct test at different logging rates.
  - Be careful as the SD card may lose connection
- When complete, unplug the USB cable and remove the SD card
- Insert the SD card into your PC and plot the data

Example Datalog

Lesson 10: Accelerometer

	A	B	C	D	E
1	Accelerometer Log				
2	########				
3	Time (ms)	X (g)	Y (g)	Z (g)	Mag (g)
4	144	-0.0625	-0.025	1.0375	1.039681
5	184	-0.075	-0.0125	1.0375	1.040282
6	224	-0.0625	-0.0125	1.025	1.02698
7	264	-0.075	0	1.025	1.02774
8	304	-0.075	0	1.0125	1.015274
9	344	-0.075	-0.0125	1.0125	1.015351
10	384	-0.075	0	1.0375	1.040207
11	424	-0.1	0	1.0375	1.042308
12	464	-0.125	-0.05	1.05	1.058596
13	504	-0.125	-0.025	1.05	1.05771
14	544	-0.125	-0.0125	1.025	1.032669
15	584	-0.1	-0.0125	1.0125	1.017503
16	624	-0.1	0	1.0125	1.017426
17	664	-0.0625	-0.025	1.0125	1.014735
18	704	-0.075	-0.0375	1.025	1.028424
19	744	-0.075	-0.0125	1.0125	1.015351
20	784	-0.075	-0.0125	1.025	1.027816
21	824	-0.075	-0.0125	1.025	1.027816
22	864	-0.0875	0	1.025	1.028728
23	904	-0.075	0	1.0125	1.015274
24	944	-0.1	0	1.025	1.029866
25	984	-0.1125	0	1.0375	1.043582
26	1024	-0.075	-0.0125	1.05	1.052749
27	1064	-0.0375	-0.0375	1.0375	1.038854
28	1104	-0.0625	-0.025	1.025	1.027208
29	1144	-0.0625	0	1.025	1.026904
30	1184	-0.075	-0.0125	1.025	1.027816
31	1224	-0.0875	0	1.025	1.028728
32	1264	-0.1	0	1.025	1.029866
33	1304	-0.075	-0.0125	1	1.002887
34	1344	-0.0625	-0.0125	1	1.002029
35	1384	-0.0875	-0.0375	1	1.004521
36	1424	-0.1125	-0.025	1.0375	1.043881
37	1464	-0.1	-0.025	1.025	1.03017

09233437

Software Settings
Accelerometer

```
myIMU.settings.adcEnabled = 0;
myIMU.settings.tempEnabled = 0; // set to 1 to enable temp sensor readings
```

– Turn the output of the accelerometer on or off (1 = on)

```
myIMU.settings.accelSampleRate = sampleRate; // Hz.Can be:0,1,10,25,50,100,200,400,1600, 5000 Hz
```

– Sensor sample rates: how fast will the sensor update its output (`0,1,10,25,50,100,200,400,1600,5000 Hz`)

```
myIMU.settings.accelRange = 16;   // 16 to read the sudden stop at the end of a drop, max G readable.
```

– Range of the accelerometer (options: 2, 4, 8, 16)
  - 2: -2g to +2g (-19.6 m/s^2 to +19.6 m/s^2)
  - 16: -16g to +16g (-156.8 m/s^2 to +156.8 m/s^2)

```
myIMU.settings.xAccelEnabled = 1;
myIMU.settings.yAccelEnabled = 1;
myIMU.settings.zAccelEnabled = 1;
```

– Turn the output of the accelerometer on or off (1 = on)

https://github.com/sparkfun/SparkFun_LIS3DH_Arduino_Library

Reading the Accelerometer

Accelerometer

`myIMU.readFloatAccelX()`

– returns a floating point number of the acceleration in g's for the X axis

`myIMU.readFloatAccelY()`

– returns a floating point number of the acceleration in g's for the Y axis

`myIMU.readFloatAccelZ()`

– returns a floating point number of the acceleration in g's for the Z axis

- More Information:
  - https://github.com/sparkfun/SparkFun_LIS3DH_Arduino_Library

Code Analysis: Dynamically creating a filename

Accelerometer

- `createFilename()`
  - Function that uses the Time from the RTC to create a unique filename each time it is called
  - Stores the filename in a global String variable called `filename`
  - Filename is formatted as DDHHMMSS.csv using the time that the file was created.
    - You can change this behavior if you need a different format say YYYYMMDD if you are creating a file every few days or so.
  - Log the data several times to see this function in action! You should find several files on your SD card.

Analysis Tools
+
Logging Trigger

Options

Analysis Tools + Logging Trigger

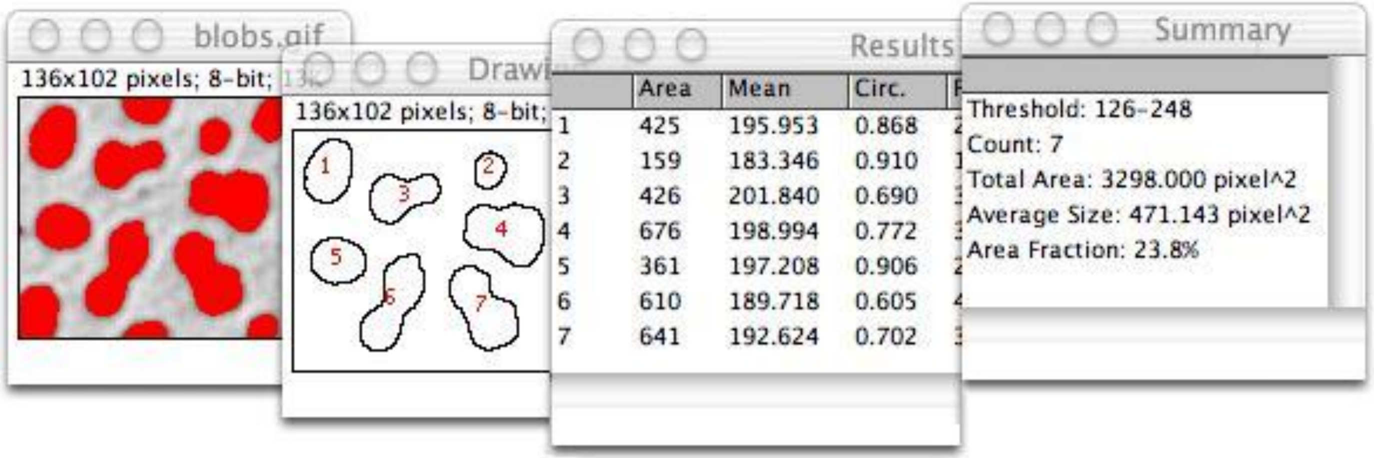
- Endless number of tools, some of the most commonly used:
  - Excel
  - Power BI
  - Matlab
  - MathCAD
  - Google Data Studio
  - Python Scripts (write your own)

- Random Other Tools
  - WinDarab (by Bosch Motorsport)

Resources

Analysis Tools + Logging Trigger

- Other Tools
  - Imagej   https://imagej.net/Welcome
    - Image processing tool



- Resources to Learn More
  - Google Analytics Academy
    - https://analytics.google.com/analytics/academy/

Triggers

Analysis Tools + Logging Trigger

- What is a Trigger?
  - A trigger starts an action when it detects an event.
  - We can program a trigger to detect a specific event and “trigger” a function of our choosing
  - We will use a trigger to detect an event such as dropping our accelerometer, and begin data capture
  - Triggers form an important and powerful part of modern test equipment data collection strategy.

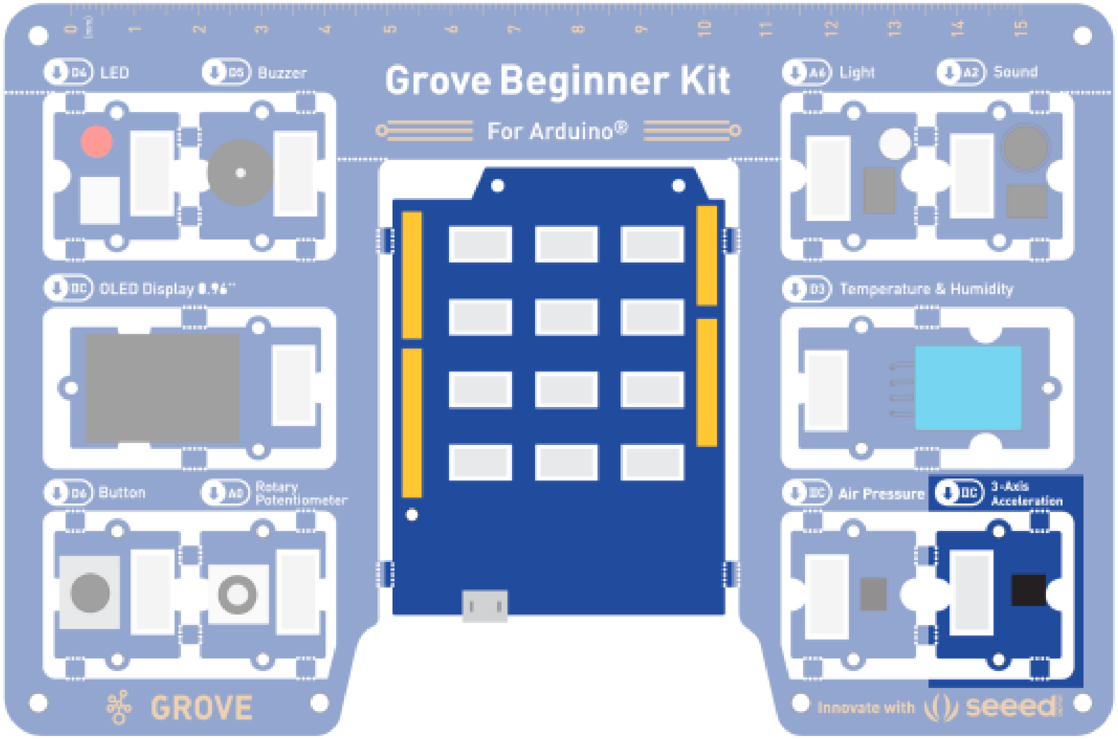
https://en.wikipedia.org/wiki/Event-driven_programming

https://en.wikipedia.org/wiki/Triggering_device

https://www.electronics-notes.com/articles/test-methods/oscilloscope/oscilloscope-trigger.php

Hardware
Accelerometer

- What hardware will we need for this Lesson?
  - Grove 3-axis Accelerometer Module on IIC
  - Seeeduino Lotus (Arduino Uno compatible board)



Open and Upload Sketch

Lesson 11: Logging Trigger *Round 1*

1. Open Simple Datalogger Sketch
   - **File → Sketchbook → FRSEF_Crash_Course → Week_5 → L11_Trigger.ino**
2. Upload the sketch to your Arduino by clicking the Upload Button.
   - The sketch should compile, and then upload to your Arduino, assuming you have the correct
3. Open the serial monitor
4. *Gently drop your board (from a couple inches)*
5. See the data stream to the serial monitor
6. Copy the data to your spreadsheet program
7. Look at the data in your spreadsheet program

Activity

Analysis Tools + Logging Trigger

- Import and plot x,y,z on a graph (use a scatter plot or line graph)
  - What data is present?
    - Beginning should show values very close to 0 for all axis
    - Then a large spike when the grove board hits a surface,
    - Secondary smaller spikes or oscillation may be seen as the board comes to rest with the magnitude around 1g

Calculating the Magnitude of Acceleration

Accelerometer

- What is the Magnitude?
  - Magnitude is the “length” or “size” of a vector.
  - For acceleration it is the amplitude of the accelerat
  - It has no direction
- How do we calculate the magnitude of our 3-
  - $Magnitude = \sqrt{X^2 + Y^2 + Z^2}$
  - Note the similarity to the Pythagorean Theorem

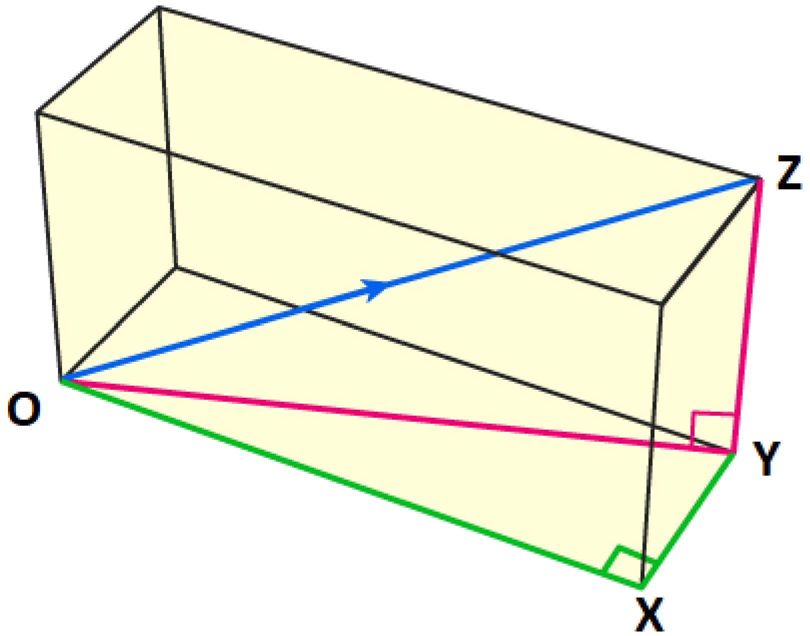


Image modified from https://byjus.com/maths/magnitude-of-a-vector/

https://en.wikipedia.org/wiki/Euclidean_vector
https://byjus.com/maths/magnitude-of-a-vector/

Code Analysis: `vectorMagnitude()` function(s)

Accelerometer

```
float aMag = vectorMagnitude(float aX, float aY, float aZ);
```

- Functions written for you and included in code to calculate the magnitude.

```
// Calculate the magnitude of a 3D vector
float vectorMagnitude(float x, float y, float z)
{
  return sqrt(x*x + y*y + z*z);
}
// Calculate the magnitude of a 2D vector
float vectorMagnitude(float x, float y)
{
  return sqrt(x*x + y*y);
}
```

Code Analysis: Drop detection

Analysis Tools + Logging Trigger

- Drop detection is done by looking for the magnitude of acceleration to be close to 0g.

```
const float fallTrigger = 0.2; // in g, less than

// Detect falling and set trigger flag
bool triggerFall(float mag)
{
  if(mag < fallTrigger)
  {
    return true; // fall or drop detected
  }
  return false; // fall or drop not detected
}
```

Code Analysis: Movement detection

Analysis Tools + Logging Trigger

- Movement detection is done by looking for the magnitude of acceleration to be significantly greater than 1g or significantly less than 1g.

```
const float moveTriggerHi = 1.2; // in g, greater than
const float moveTriggerLo = 0.8; // in g, less than

// Detect movement and set trigger flag
bool triggerMove(float mag)
{
  if((moveTriggerHi < mag) || (mag < moveTriggerLo))
  {
    return true; // movement detected
  }
  return false; // movement not detected
}
```

Code Analysis: Impact detection

Analysis Tools + Logging Trigger

- Impact detection is done by looking for the magnitude of acceleration to be much greater than 1g.

```
const float impactTrigger = 4.0; // in g, greater than

// Detect impact and set trigger flag
bool triggerImpact(float mag)
{
  if(mag > impactTrigger)
  {
    return true; // impact detected
  }
  return false; // impact not detected
}
```

Edit Sketch

Analysis Tools + Logging Trigger

- Modify the sketch so we use the pushbutton as our logging trigger
  - Original

```
if(!triggered) // if not triggered, check for trigger event
  {
    if(triggerFall(aMag) == true)
    //if(triggerMove(aMag) == true)
    //if(triggerImpact(aMag) == true)
    //if(triggerButton() == true)
```

  - New

```
if(!triggered) // if not triggered, check for trigger event
  {
    //if(triggerFall(aMag) == true)
    //if(triggerMove(aMag) == true)
    //if(triggerImpact(aMag) == true)
    if(triggerButton() == true)
```

Open and Upload Sketch

Lesson 11: Logging Trigger *Round 2*

1. Open Simple Datalogger Sketch
   – **File → Sketchbook → FRSEF_Crash_Course → Week_5 → L11_Trigger.ino**
2. Upload the sketch to your Arduino by clicking the Upload Button.
   – The sketch should compile, and then upload to your Arduino, assuming you have the correct
3. *Push the button and then gently drop your board (from a couple inches)*
4. See the data stream to the serial monitor
5. Copy the data to your spreadsheet program
6. Look at the data in your spreadsheet program

Activity

Analysis Tools + Logging Trigger

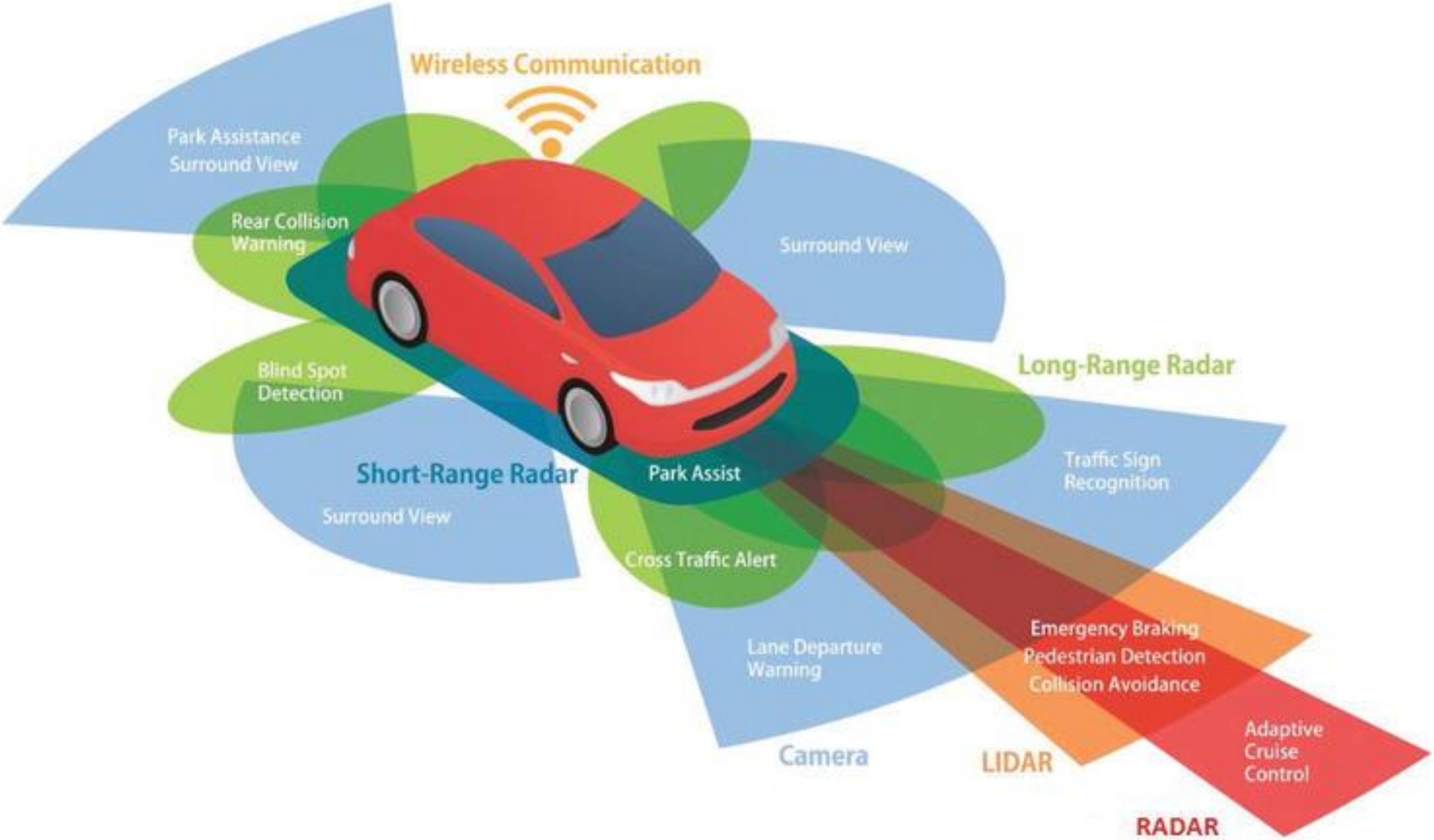
- At Home Activity
  - Modify the sketch to start recording when the board is moved
  - Modify the accelerometer settings for logging rate and time logged

Sensors & Applications

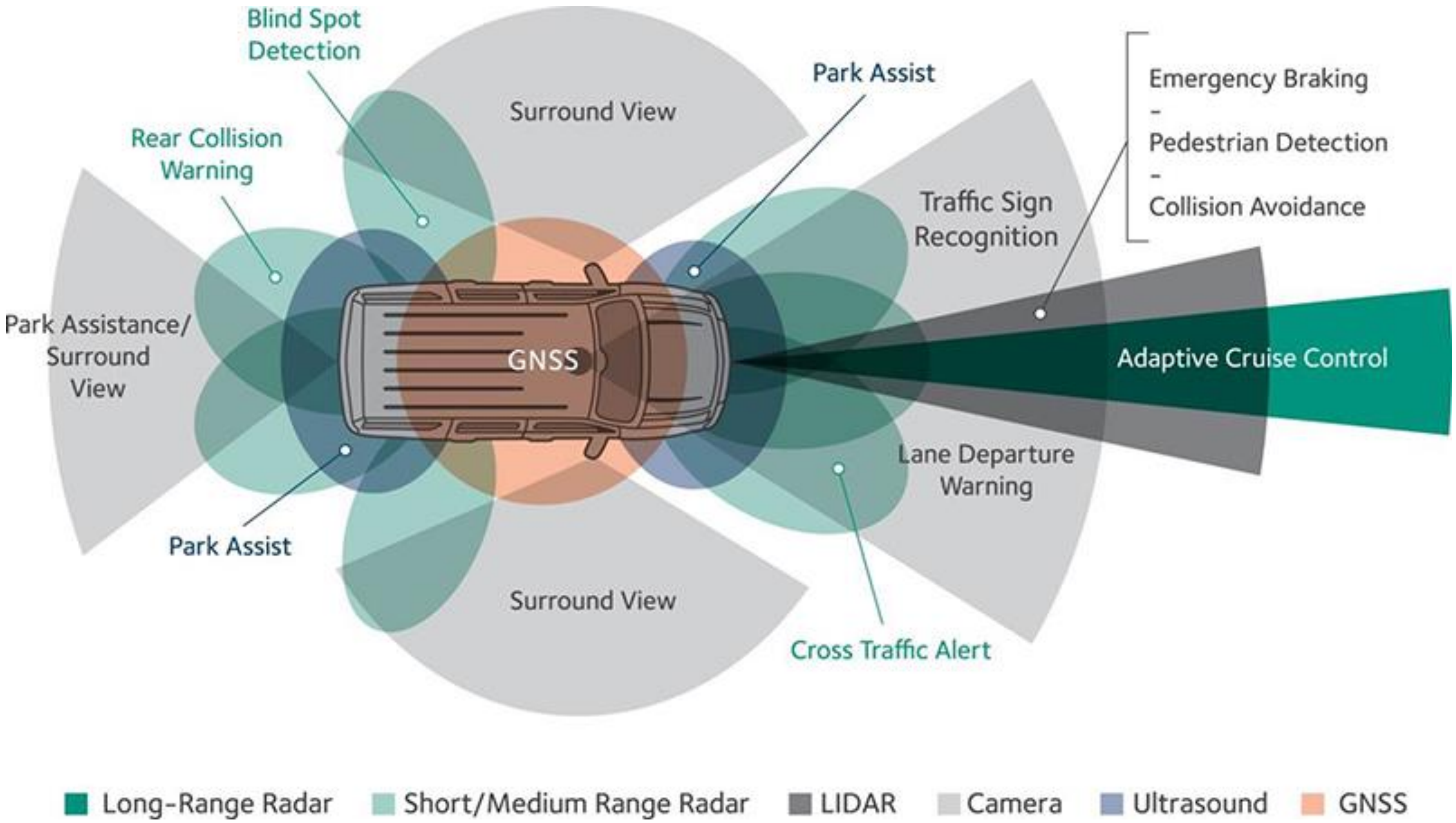
Sensors & Applications: Autonomous Driving

- Autonomous Driving
  - IMU (accel, gyro)
  - Location
    - GNSS (Global Navigation Satellite System) or GPS
  - External Object Detection
    - Radar
    - Lidar
    - Ultrasonic
    - Cameras
  - Vehicle Sensors
    - Wheel speed, steering angle

Sensors & Applications: Autonomous Driving



Sensors & Applications: Autonomous Driving



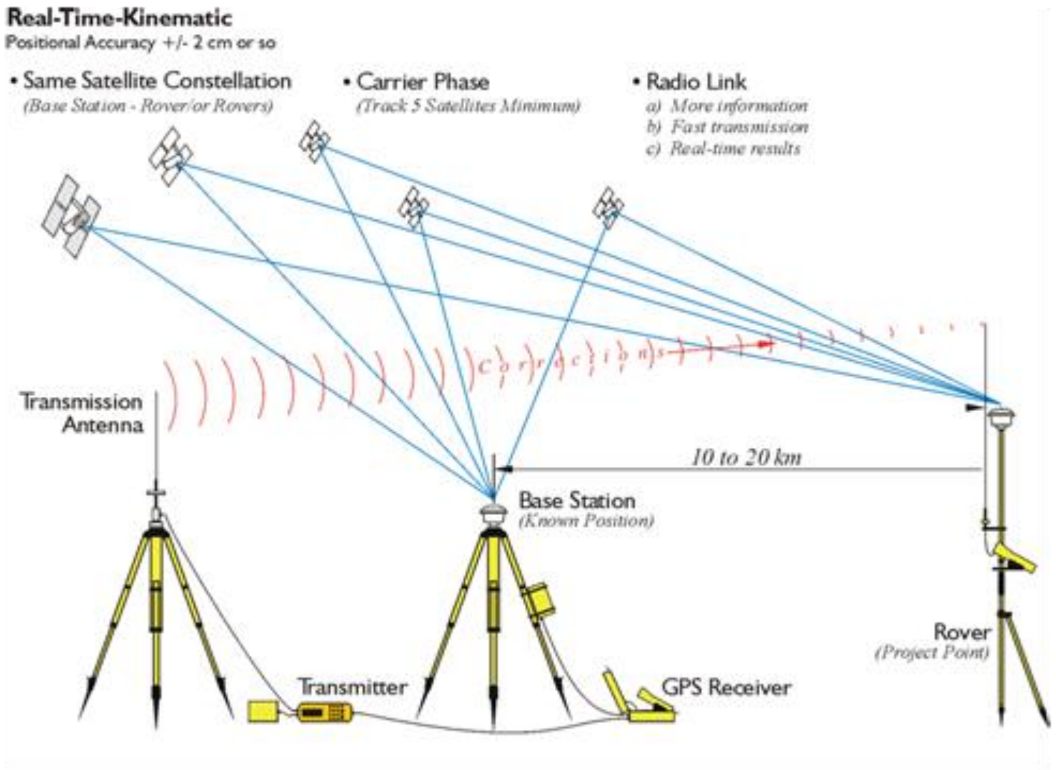
Sensors & Applications: Autonomous Driving

- GNSS (Global Navigation Satellite System)
  - Multiple systems in opreration
    - GPS (US), Galileo (EU), GLONASS (Russia), Beidou (China)
  - Time clock
  - Update rate limitations
    - Common: 10 – 20 Hz
    - Mobile Phone: 1 Hz
    - Use IMU data to calculate location between GPS updates (gps imu fusion)
    - Location Calculation: integrate acceleration to get location, small errors start to add up (drift)

GPS Update	IMU Calc	IMU Calc	IMU Calc	IMU Calc	IMU Calc	IMU Calc	IMU Calc	IMU Calc	IMU Calc	GPS Update	IMU Calc
0ms	10ms	20ms	30ms	40ms	50ms	60ms	70ms	80ms	90ms	100ms	110ms

Sensors & Applications: Autonomous Driving

- GNSS (Global Navigation Satellite System)
  - Correction services: clock drift, orbit errors, atmosphereic errors
    - PPP: Precise Point Positioning – global correction
    - RTK: Real Time Kinematic – local base station that offset data
    - DGPS: Differential GPS – base station that provides offset data



https://www.taoglas.com/centimeter-level-positioning-with-single-band-rtk/

https://www.e-education.psu.edu/geog862/node/1828

Sensors & Applications: Autonomous Driving

- Radar
  - Output radio waves and measure signals received back.
  - Short Range Radar: ~24 GHz
    - 1 to 30 m
  - Long Range Radar: ~77 GHz
    - 3 to 80 – 200m
  - Pros: Robust for weather, low cost
  - Cons: Lower resolution

Sensors & Applications: Autonomous Driving

- Radar examples

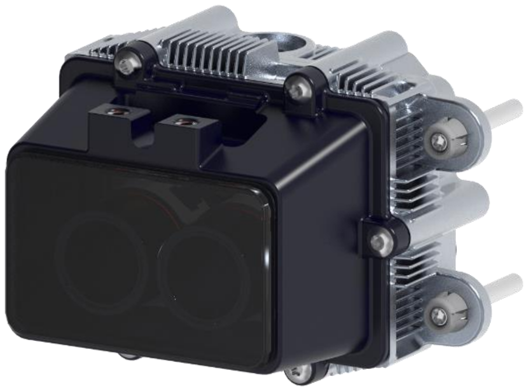
Sensors & Applications: Autonomous Driving

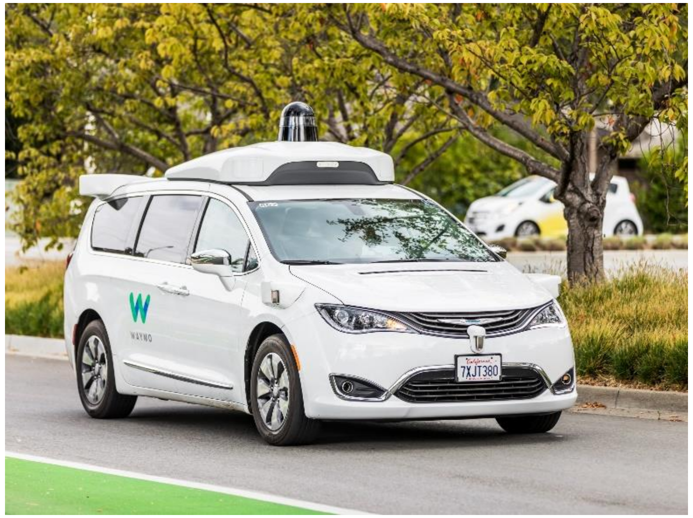
- Lidar
  - Illuminates a target with a laser and measures the characteristics of the reflected return signal.
    - Also called laser scanner, laser radar
  - Range: 0 – 200m
  - Pros: High resolution 3-D mapping, identify and classify objects
  - Cons: problems in rain, fog, snow; high cost
  - Mechanical Lidar: rotating assembly, 360° view, cost, size, robustness
    - “Orb” of google cars
  - Solid State Lidar: no spinning parts, multiple at front, rear and side combined together

https://www.ti.com/lit/wp/slyy150a/slyy150a.pdf?ts=1607565385866&ref_url=https%253A%252F%252Fwww.google.com%252F

Sensors & Applications: Autonomous Driving

- Lidar
  - Mechanical Lidar: rotating assembly, 360° view, cost, size, robustness
    - “Orb” of google cars
  - Solid State Lidar: no spinning parts, multiple at front, rear and side combined together
    - Hundreds to thousands of lasers on each module
    - iPhone 12 Pro => better photos, 3D scanning, augmented reality





Sensors & Applications: Autonomous Driving

- Camera
  - Used to detect objects and classify objects
  - Mono camera: single camera
    - Object detection and classification
  - Stereo camera: two cameras at known offset
    - Spatial awareness: measure distance and improved size calculations
    - Pros: High resolution 3-D mapping, identify and classify objects
    - Cons: High cost, high computing and software requirements